PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification: C12Q 1/68	A1	(11) International Publication Number:	WO 00/79007
		(43) International Publication Date:	28 December2000 (28.12.2000)

(21) International Application Number: PCT/US00/16899

(22) International Filing Date: 19 June 2000 (19.06.2000)

(30) Priority Data:
09/336,558 19 June 1999 (19.06.1999) US

(60) Parent Application or Grant
HYSEQ INC. [/]; (). DRMANAC, Radoje, T. [/]; (). DRMANAC, Radoje, T. [/]; (). MARSHALL, O'TOOLE, GERSTEIN, MURRAY & BORUN ; ().

Published

(54) Title: IMPROVED METHODS OF SEQUENCE ASSEMBLY IN SEQUENCING BY HYBRIDIZATION

(54) Titre: PROCEDES AMELIORES D'ASSEMBLAGE DE SEQUENCES POUR LE SEQUENÇAGE PAR HYBRIDATION

(57) Abstract

The invention relates to methods for nucleic acid sequence analysis by hybridization, in which the sequence information obtainable not only from perfectly matched oligonucleotide probes but also from oligonucleotide probes that are not perfectly matched to the target nucleic acid is taken into account.

(57) Abrégé

L'invention porte sur des procédés d'analyse séquentielle d'acides nucléiques par hybridation selon lesquels on prend en compte non seulement les séquences d'informations obtenues de sondes d'oligonucléotides parfaitement complémentaires, mais aussi, de sondes d'oligonucléotides imparfaitement complémentaires à l'acide nucléique cible.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 December 2000 (28.12.2000)

PCT

(10) International Publication Number
WO 00/79007 A1

(51) International Patent Classification[7]: C12Q 1/68

(21) International Application Number: PCT/US00/16899

(22) International Filing Date: 19 June 2000 (19.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/336,558 19 June 1999 (19.06.1999) US

(71) Applicant *(for all designated States except US)*: HYSEQ INC. [US/US]; 670 Almanor Avenue, Sunnyvale, CA 94086 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: DRMANAC, Radoje, T. [YU/US]; 850 E. Greenwich Place, Palo Alto, CA 94303 (US).

(74) Agent: MARSHALL, O'TOOLE, GERSTEIN, MURRAY & BORUN; 6300 Sears Tower, 233 South Wacker Drive, Chicago, IL 60606-6402 (US).

(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 00/79007 A1

(54) Title: IMPROVED METHODS OF SEQUENCE ASSEMBLY IN SEQUENCING BY HYBRIDIZATION

(57) Abstract: The invention relates to methods for nucleic acid sequence analysis by hybridization, in which the sequence information obtainable not only from perfectly matched oligonucleotide probes but also from oligonucleotide probes that are not perfectly matched to the target nucleic acid is taken into account.

IMPROVED METHODS OF SEQUENCE ASSEMBLY IN SEQUENCING BY HYBRIDIZATION

FIELD OF THE INVENTION

The invention relates generally to novel methods, materials and devices for nucleic acid sequence analysis by hybridization, in which the sequence information obtainable not only from perfectly matched oligonucleotide probes but also from oligonucleotide probes that are not perfectly matched to the target nucleic acid is taken into account.

BACKGROUND

The rate of determining the sequence of the four nucleotides in nucleic acid samples is a major technical obstacle for further advancement of molecular biology, medicine, and biotechnology. Nucleic acid sequencing methods which involve separation of nucleic acid molecules in a gel have been in use since 1978. The other proven method for sequencing nucleic acids is sequencing by hybridization (SBH).

The traditional method of determining a sequence of nucleotides (i.e., the order of the A, G, C and T nucleotides in a sample) is performed by preparing a mixture of randomly terminated, differentially labelled nucleic acid fragments by degradation at specific nucleotides, or by dideoxy chain termination of replicating strands. Resulting nucleic acid fragments in the range of 1 to 500 bp are then separated on a gel to produce a ladder of bands wherein the adjacent samples differ in length by one nucleotide.

The array based approach of SBH does not require single base resolution in separation, degradation, synthesis or imaging of a nucleic acid molecule. Using mismatch discriminative hybridization of short oligonucleotides K bases in length, lists of constituent K-mer oligonucleotides may be determined for target nucleic acid. Sequence for the target nucleic acid may be assembled by uniquely overlapping scored oligonucleotides.

There are several approaches available to achieve sequencing by hybridization. In a process called SBH Format 1, nucleic acid samples are arrayed, and labeled probes are hybridized with the samples. Replica membranes with the same sets of sample nucleic acids may be used for parallel scoring of several probes and/or probes may be multiplexed. Nucleic acid samples may be arrayed and hybridized on nylon membranes or other suitable supports. Each membrane array may be reused many times. Format 1 is especially efficient for batch processing large numbers of samples.

In SBH Format 2, probes are arrayed at locations on a substrate which correspond to their respective sequences, and a labelled nucleic acid sample fragment is hybridized to the arrayed probes. In this case, sequence information about a fragment may be determined in a simultaneous hybridization reaction with all of the arrayed probes. For sequencing other nucleic acid fragments, the same oligonucleotide array may be reused. The arrays may be produced by spotting or by in situ synthesis of probes.

In Format 3 SBH, two sets of probes are used. In one embodiment, a set may be in the form of arrays of probes with known positions, and another, labelled set may be stored in multiwell plates. In this case, target nucleic acid need not be labelled. Target nucleic acid and one or more labelled probes are added to the arrayed sets of probes. If one attached probe and one labelled probe both hybridize contiguously on the target nucleic acid, they are covalently ligated, producing a detected sequence equal to the sum of the length of the ligated probes. The process allows for sequencing long nucleic acid fragments, e.g. a complete bacterial genome, without nucleic acid subcloning in smaller pieces.

However, to sequence long nucleic acids unambiguously, SBH involves the use of long probes. As the length of the probes increases, so does the number of probes required to generate sequence information. Each 2-fold increase in length of the target requires a one-base increase in the length of the probe, resulting in a four-fold increase in the number of probes required (the complete set of all possible

sequences of probes of length k contains 4^k probes). For example, sequencing 100 bases of DNA requires 16,384 7-mers; sequencing 200 bases requires 65,536 8-mers; 400 bases, 262,144 9-mers; 800 bases, 1,048,576 10-mers; 1600 bases, 4,194,304 11-mers; 3200 bases, 16,777,216 12-mers; 6400 bases, 67,108,864 13-mers; and 12,800 bases requires 268,435,456 14-mers.

Because a limited number of probes can be scored in each array-based hybridization reaction, use of an extremely large number of probes requires carrying out multiple hybridization reactions.

An improvement in SBH that reduces costs and increases accuracy would greatly enhance the practical ability to sequence long pieces of polynucleotides de novo. Such an improvement would, of course, also enhance resequencing and other applications of SBH. Thus, there remains a need for additional and improved methods and materials for performing sequence analysis by hybridization.

SUMMARY OF THE INVENTION

The present invention provides novel methods and materials, including apparatus, for performing sequence analysis by hybridization (referred to herein as "SBH"). Conventional methods of SBH utilize hybridization conditions selected to discriminate probe:target hybrids that are perfectly complementary in the information region (informative region) of the probe from probe:target hybrids that have even a single base pair mismatch. Conventional methods also assemble sequence information using a scoring system for the probes that gives a "positive" score to fully matched (perfectly complementary) probes and a "negative" score to all other probes (i.e., probes with a single, double, or more base pair mismatch compared to target).

According to the present invention, the efficiency, sensitivity and accuracy of these methods is improved by taking into account the sequence information that is obtainable not only from probes that are perfectly matched to the target nucleic

acid sequence, but also from probes that have single, double or more mismatches compared to the target.

The present invention provides methods for analyzing the sequence of a target nucleic acid, comprising the steps of:

(a) contacting a target nucleic acid with a plurality of oligonucleotide probes of predetermined length and predetermined sequence, wherein each probe comprises an information region, under conditions which produce, on average, relatively more probe:target hybrids per probe for probes that are perfectly complementary in the information region of the probe than for probes that are substantially perfectly complementary in the information region of the probe, and relatively fewer probe:target hybrids that are significantly mismatched in the information region of the probe;

(b) measuring the hybridization signal of hybridization of said probes with the target nucleic acid; and

(c) assigning a numerical voting score to each probe or pool of probes based on the relative strength of hybridization signal; and

(d) determining the sequence of the target nucleic acid, comprising the step of summing the numerical voting scores of the probes in relation to their sequences.

Optionally, after step (c), the numerical voting score of the probe or pool of probes may be modified by a voting factor selected based on the relationship of the probe to the hypothetical sequence.

The number of probes used in the hybridization step may be at least about 10, at least about 100, at least about 1000, at least about 10^4, at least about 10^5, at least about 10^6, or at least about 10^7 different probes (meaning the number of distinct probe sequences), and may potentially range up to about 10^{10} different probes or even more.

The plurality of probes may be hybridized individually with the target or in groups or pools of probes. Probes may optionally be associated with or labeled

with identification tags. Each of the probes may be labeled with a unique identification tag; alternatively, a pool or a part of a pool of probes may be labeled with the same identification tag.

Another aspect of the invention provides an improvement over conventional SBH methods wherein a subpool of probes is collectively given one score, and wherein the subpool if differentiated from other subpools within the pool via labeling with distinct identification tags. Pools and subpools can be formed either with respect to probes in solution or with respect to probes in fixed arrays.

For example, a pool of 1024 pentamer probes is divided into 4 pools of 256 pentamer probes each. In each pool of 256 probes, the probes are labeled with one of four different identification tags, so that one subpool of 64 probes will bear one tag, another subpool of 64 probes will bear a second tag, a further subpool of 64 probes will bear a third tag, and a final subpool of 64 probes will bear a fourth tag. This can easily be accomplished by dividing the starting number of 1024 probes into four pools of 256 probes, labeling each pool with one of four tags, dividing the pools into four aliquots of 64 probes each, and combining an aliquot from each of the four pool to create a mixture of 256 probes labeled with 4 different tags. By creating subpools within a physical pool of probes through differential labeling of probes with identification tags, the advantages of using smaller subpools of probes can be obtained without requiring physical separation of pools into subpools. Any suitable identification tag known in the art can be used; presently preferred tags are fluorescent labels of different wavelengths or "colors." This aspect of the invention is illustrated in Figure 1. These subpools created through differential labeling can be used in the methods of the present invention, can be used as "informative pools", and can also be used in any SBH methods (including format 1, format 2 and format 3 methods) known in the art and for any SBH applications known in the art, including de novo sequencing, resequencing, polymorphism detection, etc., but are preferably utilized in resequencing applications.

Thus, this aspect of the invention provides methods of sequencing by hybridization wherein the method comprises an additional step of detecting label (identification tag). The label detection step may occur before, after, or concurrently with the steps of detecting/measuring hybridization signal.

This aspect of the invention also provides methods of sequencing by hybridization using subsets of probes that have been labeled with different tags and pooled in one or more pools by mixing all or a number of probes from each subset.

This aspect of the invention further provides a pool of probes comprising a mixture of at least 100, or at least 200, or a least 300, or at least 400 distinct probes, each probe being associated with an identification tag. The probes in the pool may all be labeled with the same tag, or may be divided into two, three, four or more subpools in which all probes in a subpool are labeled with the same tag, but each subpool is associated with a different tag. The entire subpool is given one collective score, and when sequence information is assembled it is assumed that each probe within the subpool is assigned the same score, which may be either a positive/negative score or a numerical voting score.

Another aspect of the invention provides an apparatus comprising means for carrying out the hybridization step and means for carrying out the detecting and/or measuring step(s), as described above.

A further aspect of the invention provides an apparatus comprising means for carrying out the sequence analysis step, e.g. a computer programmed as set forth in Appendix A herein.

Examples of applications that require very large numbers of probes are: (1) sequencing or resequencing of the entire human genome and other complex genomes, (2) sequencing or resequencing of total mRNA or cDNA in a human or other complex cell, (3) genotyping thousands or millions of single nucleotide polymorphisms in individual human genomes, (4) de novo sequencing of thousands of bases.

Numerous additional aspects and advantages of the invention will become apparent to those skilled in the art upon consideration of the following detailed description of the invention which describes presently preferred embodiments thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a pooling schema for the use of subpools in Format 3 sequencing by hybridization. Instead of the arrays of hexamers noted in the figure, arrays of 1024 pentamers, or arrays of 1024 pools of 4 hexamers (4096 hexamers) or arrays of 4096 pools of 4 heptamers may be used. The read length depends on the number of arrayed probes or arrayed probe pools and can range from 1 kb to several kb or more.

Figure 2 depicts the result of sequence analysis of human apo-B gene in which probes were assigned numerical voting scores based on strength of hybridization, the voting scores were further modified by a voting factor based on relationship of the probe sequence to the hypothetical target sequence, and sequence information was assembled using the informational content of both fully matched and mismatched probes.

DETAILED DESCRIPTION OF THE INVENTION

The three major steps of conventional SBH are biochemical hybridization of probes to target polynucleotide, detection of positive results, and informational sequence assembly from the results. In the biochemical hybridization step, a set of oligonucleotide probes of known sequence is allowed to hybridize with a target polynucleotide of unknown sequence. In the detection step, a subset of these probes is scored as positive (predominantly those that hybridize to and fully match the target polynucleotide sequence). In the informational step, the target sequence is assembled using different algorithms, usually executed by computer programs, that uniquely overlap all sequences of the real (or true) positive probe subset.

Conventional SBH is a well developed technology that may be practiced by a number of methods known to those skilled in the art. For example, variations of techniques related to sequencing by hybridization are described in the following documents, all of which are incorporated by reference herein: Drmanac et al., U.S. Patent No. 5,202,231 (hereby incorporated by reference herein) - Issued April 13, 1993; Drmanac et al., U.S. Patent No. 5,525,464 (hereby incorporated by reference herein) - Issued June 11, 1996; Drmanac, PCT Patent Appln. No. WO 95/09248 (hereby incorporated by reference); Drmanac et al., Genomics, 4, 114-128 (1989); Drmanac et al., Proceedings of the First Int'l. Conf. Electrophoresis Supercomputing Human Genome Cantor et al. eds, World Scientific Pub. Co., Singapore, 47-59 (1991); Drmanac et al., Science, 260, 1649-1652 (1993); Lehrach et al., Genome Analysis: Genetic and Physical Mapping, 1, 39-81 (1990), Cold Spring Harbor Laboratory Press; Drmanac et al., Nucl. Acids Res., 4691 (1986); Stevanovic et al., Gene, 79, 139 (1989); Panusku et al., Mol. Biol. Evol., 1, 607 (1990); Nizetic et al., Nucl. Acids Res., 19, 182 (1991); Drmanac et al., J. Biomol. Struct. Dyn., 5, 1085 (1991); Hoheisel et al., Mol. Gen., 4, 125-132 (1991); Strezoska et al., Proc. Nat'l. Acad. Sci. (USA), 88, 10089 (1991); Drmanac et al., Nucl. Acids Res., 19, 5839 (1991); and Drmanac et al., Int. J. Genome Res., 1, 59-79 (1992), all of which are incorporated by reference herein.

Conventional SBH approaches use arrays of target samples which are hybridized to labeled probes (Format 1), or arrays of probes which are hybridized to labeled target samples (Format 2), for efficient parallel scoring of multiple hybridization events. In one approach, either target samples or probes are attached to solid supports in the form of beads that serve to separate parallel hybridization reactions in the reading or detection step. Beads or other markers can be used as tags to identify probes. Mass spectrometry technology can also be used to distinguish probe species on the basis of their mass even when the probes are not tagged. In a Format 3 type SBH method, two sets of shorter probes are

combinatorially connected to simulate a much larger set of longer probes. Typically, in format 3, the first set of probes is fixed in the array, and the second set of probes is labeled. The labeled probes are added together with target nucleic acid, and a labeled probe is ligated to a fixed probe only if the two probes hybridize contiguously on the target nucleic acid.

Format 1, 2 and 3 SBH methods are described in further detail below. In addition, a set of probes can be scored in the form of informative pools with minimal loss of information, as described in U.S. Application Serial No. 60/115,284 entitled "Enhanced Sequencing by Hybridization Using Informative Pools of Probes" filed January 6, 1999, incorporated herein by reference. Other types of pools may be used. In addition to pooling probes that cannot be distinguished in the reading step, probes with unique tags can be multiplexed in the same hybridization reaction. Probes can be individually synthesized in separate reactions or in situ, or a combination of two much smaller sets of shorter probes may be used to score a much larger set of longer probes (1024 5-mers x 1024 5-mers = 1,048,576 10-mers).

Unlike conventional SBH, which only uses and assembles sequence information from full match probes (*i.e.*, probes that are perfectly matched to the target nucleic acid), the present invention utilizes hybridization information obtainable not only from probes that are perfectly matched to the target nucleic acid sequence, but also from probes that have single, double or more mismatches compared to the target.

There is typically a wide spread, or range, of hybridization (or hybridization/ligation) signals for full match probe:target hybrids, due to differences in hybrid stability, target accessibility and variations in probe quality and quantity. In conventional SBH, a single threshold is set for a positive score, meaning that probes with hybridization signals higher than the threshold are scored as positive and probes with hybridization signals lower than the threshold are scored as negative. Conventional procedures and algorithms for base calling

(determining the identity of the nucleotide at a particular position), either for resequencing using known reference sequences or for de novo sequencing, use this type of positive/negative data to maximize accuracy and read length.

However, there is also a wide range of signals for probe:target hybrids containing a single base pair mismatch, depending on the location of the mismatch and the type of mismatch, and there is a wide range of signals for probe:target hybrids containing a double base pair mismatch, etc. Consequently, the range of signals for full match probe:target hybrids often overlaps the range of signals for mismatched probe:target hybrids. In other words, the lower part of the range of signals for full match probe:target hybrids is often mixed with strong signals from mismatched probe:target hybrids (usually single mismatches).

Thus, the setting of a single threshold may have the undesirable effects of creating false positive probes (*i.e.*, probes that form strongly hybridizing single mismatch probe:target hybrids) and false negative probes (*i.e.*, probes that form weakly hybridizing full match probe:target hybrids). In addition, a system that scores only full match probe:target hybrids ignores the useful sequence information that can be provided by mismatched probe:target hybrids, particularly single mismatches. For example, if 10-mer probes are being used, for all of the single mismatch probes, 9 of the 10 bases will be a correct identification of the true base at that position. The numerical scoring of all probes according to the relative level or strength of hybridization allows the informational content of all probes to be taken into account.

The present invention provides methods for analyzing the sequence of a target nucleic acid, comprising the steps of:

(a) contacting a target nucleic acid with a plurality of oligonucleotide probes of predetermined length and predetermined sequence, wherein each probe comprises an information region, under conditions which produce, on average, relatively more probe:target hybrids per probe for probes that are perfectly complementary in the information region of the probe than for probes that are

substantially perfectly complementary in the information region of the probe, and relatively fewer probe:target hybrids that are significantly mismatched in the information region of the probe;

(b) measuring the hybridization signal of said probe:target hybrids; and

(c) assigning a numerical voting score to each probe or pool of probes based on the relative strength of hybridization signal; and

(d) determining the sequence of the target nucleic acid, comprising the step of analyzing the numerical voting scores of the probes in relation to their sequences.

In step (a), the hybridization and/or wash conditions are selected to produce, on average, a relatively higher number of probe:target hybrids for fully matched (perfectly complementary) probes than for probes with a single mismatch compared to target, and a relatively higher number of hybrids for single mismatch probes compared to double mismatch probes, and a relatively higher number of hybrids for double mismatch probes compared to triple mismatch probes, etc.

In step (b), the hybridization signal of the probe:target hybrids is measured and the relative level (or strength) of the signal is determined.

In step (c), a numerical voting score (also referred to herein as "voting power") is assigned to each probe or to a pool of probes based on the strength of the hybridization signal. The assignment of scores is described below in more detail in the section entitled "Assigning a Numerical Voting Score to Probes and Sequence Analysis."

In step (d), the sequence may be analyzed by aligning all possible sequences, summing the numerical voting scores of all probes voting for a particular hypothetical base identity at a particular position, and determining which of the hypothesized bases is correct (*i.e.*, has the most votes). Alternatively, the numerical voting score of the probes as assigned in step (c) may be further modified, or weighted, by a voting factor as described in the section entitled "Assigning a Numerical Voting Score to Probes and Sequence Analysis," wherein

the modification of the score for a probe depends on the relationship of that probe to the hypothesized sequence.

The advantages of this approach to sequence assembly are that it eliminates the need for the separation of full and mismatched probes and thus reduces the number of false positive and false negatives, it obtains maximal benefit from the informational content of probes that form single or double mismatches, and it uses the majority of probes with readable matches, thus increasing the redundancy of reads several-fold compared to using full match probes only.

Target Polynucleotide

"Target nucleic acid" or "target polynucleotide" refers to the nucleic acid of interest, typically the nucleic acid that is sequenced in the SBH assay. Potential target polynucleotides include naturally occurring or artificially created DNA (*e.g.*, genomic DNA and cDNA) and RNA (*e.g.*, mRNA), including nucleic acids used as part of DNA computing. The target nucleic acid may be composed of ribonucleotides, deoxyribonucleotides or mixtures thereof. Typically, the target nucleic acid is a DNA. While the target nucleic acid can be double-stranded, it is preferably single stranded. The "read length" of the target nucleic acid can be any number of nucleotides, depending on the length of the probes, but is typically on the order of 100, 200, 400, 800, 1600, 3200, 6400, or even more nucleotides in length, up to the entire human genome.

The target nucleic acid can be obtained from virtually any source and can be prepared using methods known in the art. For example, target nucleic acids can be isolated by PCR methodology, or by cloning into plasmids (for a convenient target nucleic acid fragment length of 500 to 5,000 base pairs), or by cloning into yeast or bacterial artificial chromosomes (for a convenient target nucleic acid fragment length of up to 100kb).

Depending on the desired length for use in the SBH assay, the target nucleic acid may be sheared into fragments prior to use in an SBH assay.

Fragmentation may be accomplished by nonspecific endonuclease digestion, restriction enzyme digestion (*e.g.*, by *Cvi JI*), physical shearing (*e.g.*, by ultrasound) or NaOH treatment. Fragments may be separated by size (*e.g.*, by gel electrophoresis) to obtain the desired fragment length. Fragmentation of the target nucleic acid also may avoid hindrance to hybridization from secondary structure in the sample. The sizes of the target nucleic acid fragments used in the hybridization reaction optimally range in length from slightly longer than the probe length to twice the probe length, *e.g.*, 10-100 or 10-40 bases.

Probes

"Probes" refers to relatively short pieces of nucleic acids, preferably DNA. Probes are preferably shorter than the target DNA by at least one base, and more preferably they are 25 bases or fewer in length, still more preferably 20 bases or fewer in length. Of course, the optimal length of a probe will depend on the length of the target nucleic acid being analyzed. For a target nucleic composed of about 100 or fewer bases, the probes are at least 7-mers; for a target of about 100-200 bases, the probes are at least 8-mers; for a target nucleic acid of about 200-400 bases, the probes are at least 9-mers; for a target nucleic acid of about 400-800 bases, the probes are at least 10-mers; for a target nucleic acid of about 800-1600 bases, the probes are at least 11-mers; for a target of about 1600-3200 bases, the probes are at least 12-mers, for a target of about 3200-6400 bases, the probes are at least 13-mers; and for a target of about 6400-12,800 bases, the probes are at least 14-mers. For every additional two-fold increase in the length of the target nucleic acid, the optimal probe length is one additional base. Those of skill in the art will recognize that for Format 3 SBH applications, the above-delineated probe lengths are post-ligation. Thus, as used throughout, specific probe lengths refer to the actual length of the probes for Format 1 and 2 SBH applications and the lengths of ligated probes for Format 3 or Format 3-like SBH applications. Probes

are normally single stranded, although double-stranded probes may be used in some applications.

Probes may be prepared using standard chemistry procedures known in the art. The length of the probes described above refers to the length of the informational content of the probes, not necessarily the actual physical length of the probes. The probes used in SBH frequently contain degenerate ends [*e.g.*, one to three non-specified (mixed A,T,C and G) or universal (e.g. M base or inosine) bases at the ends] that aid hybridization but do not contribute to the information content of the probes. Hybridization discrimination of mismatches in these degenerate probe mixtures refers only to the length of the informational content, not the full physical length. For example, SBH applications frequently use mixtures of probes of the formula NxByNz, wherein N represents any of the four bases and varies for the polynucleotides in a given mixture, B represents any of the four bases but is the same for each of the polynucleotides in a given mixture, and x, y, and z are all integers. In this formula, Nx and Nz represent the degenerate ends of the probe and By represents the information content of the probe (*e.g.*, a uniquely arrayed probe in conventional SBH).

The probes may consist solely of naturally-occurring nucleotides and native phosphodiester backbones, or the probes may be modified or tagged to enhance specificity of detection. For example, the probes may be composed of one or more modified bases, such as 7-deazaguanosine, or one or more modified backbone interlinkages, such as a phosphorothioate. The only requirement is that the probes be able to hybridize to the target nucleic acid. A wide variety of modified bases and backbone interlinkages that can be used in conjunction with the present invention are known, and will be apparent to those of skill in the art.

Other variations include the use of modified oligonucleotides to increase specificity or efficiency, cycling hybridizations to increase the hybridization signal, for example by performing a hybridization cycle under conditions (e.g. temperature) optimally selected for a first set of labeled probes followed by

hybridization under conditions optimally selected for a second set of labeled probes. Shifts in reading frame may be determined by using mixtures (preferably mixtures of equimolar amounts) of probes ending in each of the four nucleotide bases A, T, C and G.

The oligonucleotide probes are preferably labeled with identification tags to enhance detection or discrimination. Suitable labels include fluorescent dyes, chemiluminescent systems, radioactive labels (e.g., ^{35}S, ^{3}H, ^{32}P or ^{33}P), or isotopes detectable by mass spectrometry, nanobeads, polymers or molecules of different size, shape, electrical, magnetic or other properties, attached by any of a variety of methods that are well known in the art.

A complete set of all possible probes of a given length (4^N, where N is the length) or a subset of this complete set may be used in the hybridization step. Probes of differing lengths may also be used. A large number of probes may be synthesized in a small number of reactions. For example, a complete set of all possible 10-mers (about 1 million probes) may be synthesized as follows. 1000 5-mers, each uniquely associated with a 10-digit DNA bar code, are synthesized in 1000 reactions and mixed. The mixture is divided into 1000 aliquots which then undergo 1000 reactions, during which the informational length of the probe is extended by an additional 5 nucleotides and the 10-digit barcode is extended by a further 10 digits, to form 1 million uniquely tagged 10-mers synthesized in only 2000 reactions.

Hybridization Reaction

The number and type of probes that are used in each hybridization reaction depends on the detection power of the reader, the statistics of numerical scoring, including use of informative pools (or other pools), the length of target nucleic acid sequence, and the SBH application (*e.g.*, whether de novo sequencing, resequencing or genotyping is desired). A complete set of all possible probe

sequences of the same length may be used, or only a portion of this complete set may be used. Alternatively, probes of differing length may be used.

Hybridization Conditions

Hybridization and washing conditions are selected to provide a range of hybridization signals such that a gradation of signals is provided wherein full match probes have higher signals than single mismatch probes, which in turn have higher signals than double mismatch probes, which in turn have higher signals than triple mismatch probes, etc. Conditions may be selected so as to detect substantially perfect match hybrids (such as those wherein the fragment and probe hybridize at six out of seven positions). Alternatively, slightly less stringent conditions than those that permit detection only of perfect match hybrids may be used.

Suitable hybridization conditions may be routinely determined by optimization procedures or pilot studies. Such procedures and studies are routinely conducted by those skilled in the art to establish protocols for use in a laboratory. See e.g., Ausubel et al., Current Protocols in Molecular Biology, Vol. 1-2, John Wiley & Sons (1989); Sambrook et al., Molecular Cloning A Laboratory Manual, 2nd Ed., Vols. 1-3, Cold Springs Harbor Press (1989); and Maniatis et al., Molecular Cloning: A Laboratory Manual, Cold Spring Harbor Laboratory Cold Spring Harbor, New York (1982), all of which are incorporated by reference herein. For example, conditions such as temperature, concentration of components, hybridization and washing times, buffer components, and their pH and ionic strength may be varied.

Assigning a Numerical Voting Score to Probes and Sequence Analysis

The probes which have hybridized to the target polynucleotide during the hybridization reaction step can be assigned a numerical voting score (voting power) based on the strength of the hybridization signal.

Data may be obtained by scoring each probe individually or by scoring pools of probes, including informative pools as described in U.S. Application Serial No. 60/115,284 entitled "Enhanced Sequencing by Hybridization Using Informative Pools of Probes" filed January 6, 1999, incorporated herein by reference.

Probes can numerically scored and their sequences analyzed as follows. Probes are sorted by descending hybridization signal value, and a numerical voting score (voting power) is assigned to each probe based on the hybridization signal and a converting function. One possible converting function involves dividing the signal range into several segments (by setting a certain number of threshold steps) and to define the voting power for probes in each segment as the inverse of the number of probes in that segment. The lower boundary of the signal range that is taken into account is set at least at the background level of signal, but may be set higher as desired in order to simplify or speed computation without losing significant information.

Probe sequences are aligned allowing for mismatches. For base calling, for example in resequencing, the identity of a base at a particular position is voted on by the number of occurrences of a base in aligned probes in combination with the numerical voting score (or voting power) of each probe. The numerical voting scores of all probes voting for a particular hypothetical base identity at a particular position may be summed, and the base identity may be confirmed by determining which of the hypothesized bases has the most votes. For example, because the voting power of a strongly hybridizing probe is higher, its vote as to the identity of the base is given more weight.

Alternatively, the assigned numerical voting score of the probes may be further modified, or weighted, by a voting factor, wherein the modification of the score for a probe depends on the relationship of that probe to the hypothesized sequence. For example, when 10-mers are used in resequencing, if the base at position 100 is hypothesized to be an adenine (A), there will be 10 full match

(perfectly complementary) probes for the A at that position, and 270 (9 x 3 x 10) single mismatch probes for the A that position. The voting factor by which the numerical voting score is modified may be set to a multiplier of 100 for full match probes, a multiplier of 20 for single mismatch probes, and a multiplier of 2 for all other probes. The votes are then summed after modification by the appropriate voting factor, and the voting process is repeated for each of the four hypothesized bases at position 100 (A, T, C and G). The hypothesized base that has the highest number of votes may be declared the correct base. This voting process can be used to solve single-position base problems or can be used to determine the identity of two consecutive base positions (in which case there are sixteen, rather than four, hypothesized two-base combinations).

However, as illustrated in Example 1 below, a correct solution requires an absolute minimum number of votes. If none of the hypothetical base candidates receives the minimum number of votes, then the process needs to be repeated. In addition, a correct solution requires the "winning" candidate to have a sufficiently high number of votes in comparison to the other candidates. In the case of a heterozygous position (two genes are present, and each gene has a different base at that position), there can be two "winning" candidates, but each of the "winners" must still have a sufficiently high number of votes compared to the other candidates.

Alternatively, the voting power of probes can be used to determine whether a probe is a full match probe, for example, in de novo sequencing. The selected probe in question is aligned with all probes that have a single or more mismatches (mismatches when compared to the probe in question). Statistics are applied that take into account the voting power of each of the mismatched probes (and may include the vote of the selected probe itself). If the selected probe is actually a full match probe, probes that have a single mismatch compared to the selected probes should still hybridize strongly. However, if the selected probe is actually a strongly hybridizing single mismatched probe, probes that have a single mismatch compared

to the selected probe will be double mismatched probes compared to the target nucleic acid sequence and thus will hybridize relatively more weakly. For example, summing the numerical voting score (or the numerical voting score as modified by voting factor) of all probes having a single mismatch in comparison to the selected probe thus will indicate whether the selected probe is a full match.

Probes with end mismatches typically have a stronger hybridization signal than probes with internal mismatches and thus it may be desirable to set voting factors so that these probes are given relatively more voting power.

For determining the correct full match probes among several closely related probes that share an identical middle portion (especially important in de novo sequencing), 5' to 3' positions of mismatch may be taken into account. For example, bridging across a branching point of a repeated 6-mer sequence can be done by sorting all probes by the central 6-mer and determining which of the positive probes are full match probes. If, for example, there are 6 positive probes sharing the same middle 6-mer sequence, and one assumes that only two probes are true full match probes, then single and double mismatches may be taken into account when voting for the full match probes. This approach has the advantages of eliminating false positives and reducing the occurrence of false sequence assembly.

Example 1

The above described sequence analysis methods of the invention were carried out for resequencing of a 700 base pair fragment of the human apo-B gene in one cardiovascular patient as follows.

DNA was prepared by PCR with one phosphorylated primer. Lambda exonuclease was used to degrade the phosphorylated strand and the remaining single stranded DNA was randomly fragmented by endonuclease DNAse I. The target DNA was mixed with 16 pools containing 64 TAMRA labeled 5-mer probes and hybridized to 4 HyChips each containing four 5-mer arrays. The hybridization

image was detected using a fluorescent scanner and a hybridization score for each of about 16,000 test dots was determined using an image analysis program.

Probes were sorted by descending hybridization signal value, and a numerical voting score was assigned to each probe based on its hybridization signal and a converting function. In this case, for the top 2000 dots (each corresponding to a pool of 64 pentamers scoring 64 10-mers), the probes were assigned an initial numerical voting score that was equal to their hybridization signal, while all other probes were assigned a numerical voting score of zero.

The assigned numerical voting score of each probe was further weighted by a voting factor. The voting factor was set to a multiplier of 100 for full match probes and a multiplier of 1 for all other probes. The votes were then summed after modification by the appropriate voting factor. The modified numerical voting scores (modified by the appropriate voting factor) of all probes voting for a particular hypothetical base identity at a particular position were summed, and the base identity was confirmed by determining which of the hypothesized bases had the most votes.

The results of the sequence analysis according to this voting schema are shown in Figure 2. The figure depicts a 400-500 base segment of the 700 base pair fragment sequenced. For each base position, all four nucleotide options were tested. The sum of the votes is plotted for each nucleotide at each position, and the points in the graph are marked with corresponding nucleotide letters. The capital letters denote the apo-B reference sequence. At positions 485 and 486, the sequence is undeterminable because none of the four bases received a minimum number of votes (each base received a total score of approximately 2000 votes each). At position 471, there are two correct bases (G and A) that each received a total score of approximately 10,000 (a 10-fold difference relative to the other two base candidates), indicating that the sample is heterozygous.

Description of conventional Format 1 and 2 SBH

A. Assay format

Format 1 SBH is appropriate for the simultaneous analysis of a large set of samples. Parallel scoring of thousands of samples on large arrays may be performed in thousands of independent hybridization reactions using small pieces of membranes. The identification of DNA may involve 1-20 probes per reaction and the identification of mutations may in some cases involve more than 1000 probes specifically selected or designed for each sample. For identification of the nature of the mutated DNA segments, specific probes may be synthesized or selected for each mutation detected in the first round of hybridizations.

DNA samples may be prepared in small arrays which may be separated by appropriate spacers, and which may be simultaneously tested with probes selected from a set of oligonucleotides which may be arrayed in multiwell plates. Small arrays may consist of one or more samples. DNA samples in each small array may include mutants or individual samples of a sequence. Consecutive small arrays may be organized into larger arrays. Such larger arrays may include replication of the same small array or may include arrays of samples of different DNA fragments. A universal set of probes includes sufficient probes to analyze a DNA fragment with prespecified precision, e.g. with respect to the redundancy of reading each base pair ("bp"). These sets may include more probes than are necessary for one specific fragment, but may include fewer probes than are necessary for testing thousands of DNA samples of different sequence.

DNA or allele identification and a diagnostic sequencing process may include the steps of:

1) Selection of a subset of probes from a dedicated, representative or universal set to be hybridized with each of a plurality of small arrays;

2) Adding a first probe to each subarray on each of the arrays to be analyzed in parallel;

3) Performing hybridization and scoring of the hybridization results;

4) Stripping off previously used probes;

5) Repeating hybridization, scoring and stripping steps for the remaining probes which are to be scored;

5) Processing the obtained results to obtain a final analysis or to determine additional probes to be hybridized;

6) Performing additional hybridizations for certain subarrays; and

7) Processing complete sets of data and computing obtaining a final analysis.

This approach provides fast identification and sequencing of a small number of nucleic acid samples of one type (e.g. DNA, RNA), and also provides parallel analysis of many sample types in the form of subarrays by using a presynthesized set of probes of manageable size. Two approaches have been combined to produce an efficient and versatile process for the determination of DNA identity, for DNA diagnostics, and for identification of mutations.

For the identification of known sequences, a small set of shorter probes may be used in place of a longer unique probe. In this approach, although there may be more probes to be scored, a universal set of probes may be synthesized to cover any type of sequence. For example, a full set of 6-mers includes only 4,096 probes, and a complete set of 7-mers includes only 16,384 probes.

Full sequencing of a DNA fragment may be performed with two levels of hybridization. One level is hybridization of a sufficient set of probes that cover every base at least once. For this purpose, a specific set of probes may be synthesized for a standard sample. The results of hybridization with such a set of probes reveal whether and where mutations (differences) occur in non-standard samples. To determine the identity of the changes, additional specific probes may be hybridized to the sample.

In another variation, all probes from a universal set may be scored. A universal set of probes allows scoring of a relatively small number of probes per sample in a two step process without an undesirable expenditure of time. The

hybridization process may involve successive probings, in a first step of computing an optimal subset of probes to be hybridized first and, then, on the basis of the obtained results, a second step of determining additional probes to be scored from among those in a universal set.

B. Sequence Assembly

In SBH sequence assembly, K -1 oligonucleotides which occur repeatedly in analyzed DNA fragments due to chance or biological reasons may be subject to special consideration. If there is no additional information, relatively small fragments of DNA may be fully assembled in as much as every base pair is read several times.

In the assembly of relatively longer fragments, ambiguities may arise due to the repeated occurrence in a set of positively-scored probes of a K-1 sequence (i.e., a sequence shorter than the length of the probe). This difficulty does not exist if mutated or similar sequences have to be determined. Knowledge of one sequence may be used as a template to correctly assemble a sequence known to be similar (e.g. by its presence in a database) by arraying the positive probes for the unknown sequence to display the best fit on the template.

Within DNA, the location of certain probes may be interchangeable when determined by overlapping the sequence data, resulting in an ambiguity as to the position of the partial sequence. Although the sequence information is determined by SBH, either: (i) long read length, single-pass gel sequencing at a fraction of the cost of complete gel sequencing; or (ii) comparison to related sequences, may be used to order hybridization data where such ambiguities ("branch points") occur. In addition, segments in junk DNA (which is not found in genes) may be repeated many times in tandem. Although the sequence of the segments is determined by SBH, single-pass gel sequencing may be used to determine the number of tandem repeats where tandemly-repeated segments occur. As tandem repeats occur rarely

in protein-encoding portions of a gene, the gel-sequencing step will be performed only when a commercial value for the sequence is determined.

C. Sequencing of Mutants

The use of an array of sample arrays avoids consecutive scoring of many oligonucleotides on a single sample or on a small set of samples. This approach allows the scoring of more probes in parallel by manipulation of only one physical object. Subarrays of DNA samples 1000 bp in length may be sequenced in a relatively short period of time. If the samples are spotted at 50 subarrays in an array and the array is reprobed 10 times, 500 probes may be scored. In screening for the occurrence of a mutation, approximately 335 probes may be used to cover each base three times. If a mutation is present, several covering probes will be affected. The use of information about the identity of negative probes may map the mutation with a two base precision. To solve a single base mutation mapped with this precision, an additional 15 probes may be employed. These probes cover any base combination for two questionable positions (assuming that deletions and insertions are not involved). These probes may be scored in one cycle on 50 subarrays which contain a given sample. In the implementation of a multiple label color scheme (i.e., multiplexing), two to six probes, each having a different label such as a different fluorescent dye, may be used as a pool, thereby reducing the number of hybridization cycles and shortening the sequencing process.

In more complicated cases, there may be two close mutations or insertions. They may be handled with more probes. For example, a three base insertion may be solved with 64 probes. The most complicated cases may be approached by several steps of hybridization, and the selecting of a new set of probes on the basis of results of previous hybridizations.

If subarrays to be analyzed include tens or hundreds of samples of one type, then several of them may be found to contain one or more changes (mutations, insertions, or deletions). For each segment where mutation occurs, a specific set of

probes may be scored. The total number of probes to be scored for a type of sample may be several hundreds. The scoring of replica arrays in parallel facilitates scoring of hundreds of probes in a relatively small number of cycles. In addition, compatible probes may be pooled. Positive hybridizations may be assigned to the probes selected to check particular DNA segments because these segments usually differ in 75% of their constituent bases.

By using a larger set of longer probes, longer targets may be conveniently analyzed. These targets may represent pools of shorter fragments such as pools of exon clones.

D. Identification of Heterozygotes Using SBH

A specific hybridization scoring method may be employed to define the presence of heterozygotes (sequence variants) in a genomic segment to be sequenced from a diploid chromosomal set. Two variations are where: i) the sequence from one chromosome represents a basic type and the sequence from the other represents a new variant; or, ii) both chromosomes contain new, but different variants. In the first case, the scanning step designed to map changes gives a maximal signal difference of two-fold at the heterozygotic position. In the second case, there is no masking, but a more complicated selection of the probes for the subsequent rounds of hybridizations may be indicated.

Scoring two-fold signal differences required in the first case may be achieved efficiently by comparing corresponding signals with controls containing only the basic sequence type and with the signals from other analyzed samples. This approach allows determination of a relative reduction in the hybridization signal for each particular probe in a given sample. This is significant because hybridization efficiency may vary more than two-fold for a particular probe hybridized with different DNA fragments having the same full match target. In addition, heterozygotic sites may affect more than one probe depending upon the number of oligonucleotide probes. Decrease of the signal for two to four

consecutive probes produces a more significant indication of heterozygotic sites. Results may be checked by testing with small sets of selected probes among which one or few probes selected to give a full match signal which is on average eight-fold stronger than the signals coming from mismatch-containing duplexes.

Partitioned membranes allow a very flexible organization of experiments to accommodate relatively larger numbers of samples representing a given sequence type, or many different types of samples represented with relatively small numbers of samples. A range of 4-256 samples can be handled with particular efficiency. Subarrays within this range of numbers of dots may be designed to match the configuration and size of standard multiwell plates used for storing and labeling oligonucleotides. The size of the subarrays may be adjusted for different number of samples, or a few standard subarray sizes may be used. If all samples of a type do not fit in one subarray, additional subarrays or membranes may be used and processed with the same probes. In addition, by adjusting the number of replicas for each subarray, the time for completion of identification or sequencing process may be varied.

Signature Analysis with SBH

Obtaining information about the degree of hybridization exhibited for a set of only about 200 oligonucleotides probes (about 5% of the effort required for complete sequencing) defines a unique signature of each gene and may be used for sorting the cDNAs from a library to determine if the library contains multiple copies of the same gene. By such signatures, identical, similar and different cDNAs can be distinguished and inventoried.

Format 3 Sequencing by Hybridization

In Format 3, a first set of oligonucleotide probes of known sequence is immobilized on a solid support under conditions which permit them to hybridize with nucleic acids having respectively complementary sequences. A labeled,

second set of oligonucleotide probes is provided in solution. Both within the sets and between the sets the probes may be of the same length or of different lengths. A nucleic acid to be sequenced or intermediate fragments thereof may be applied to the first set of probes in double-stranded form (especially where a recA protein is present to permit hybridization under non-denaturing conditions), or in single-stranded form and under conditions which permit hybrids of different degrees of complementarity (for example, under conditions which discriminate between full match and one base pair mismatch hybrids). The nucleic acid to be sequenced or intermediate fragments thereof may be applied to the first set of probes before, after or simultaneously with the second set of probes. A ligase or other means of causing chemical bond formation between adjacent, but not between nonadjacent, probes may be applied before, after or simultaneously with the second set of probes. After permitting adjacent probes to be chemically bonded, fragments and probes which are not immobilized to the surface by chemical bonding to a member of the first set of probe are washed away, for example, using a high temperature (up to 100 degrees C) wash solution which melts hybrids. The bound probes from the second set may then be detected using means appropriate to the label employed (which may be chemiluminescent, fluorescent, radioactive, enzymatic or densitometric, for example).

Herein, nucleotide bases "match" or are "complementary" if they form a stable duplex by hydrogen bonding under specified conditions. For example. under conditions commonly employed in hybridization assays, adenine ("A") matches thymine ("T"), but not guanine ("G") or cytosine ("C"). Similarly, G matches C, but not A or T. Other bases which will hydrogen bond in less specific fashion, such as inosine or the Universal Base ("M" base, Nichols et al 1994), or other modified bases, such as methylated bases, for example, are complementary to those bases for which they form a stable duplex under specified conditions. A probe is said to be "perfectly complementary" or is said to be a "perfectly match" if each base in the probe forms a duplex by hydrogen bonding to a base in the nucleic acid to be

sequenced. Each base in a probe that does not form a stable duplex is said to be a "mismatch" under the specified hybridization conditions.

A list of probes may be assembled wherein each probe is a perfect match to the nucleic acid to be sequenced. The probes on this list may then be analyzed to order them in maximal overlap fashion. Such ordering may be accomplished by comparing a first probe to each of the other probes on the list to determine which probe has a 3' end which has the longest sequence of bases identical to the sequence of bases at the 5' end of a second probe. The first and second probes may then be overlapped, and the process may be repeated by comparing the 5' end of the second probe to the 3' end of all of the remaining probes and by comparing the 3' end of the first probe with the 5' end of all of the remaining probes. The process may be continued until there are no probes on the list which have not been overlapped with other probes. Alternatively, more than one probe may be selected from the list of positive probes, and more than one set of overlapped probes ("sequence nucleus") may be generated in parallel. The list of probes for either such process of sequence assembly may be the list of all probes which are perfectly complementary to the nucleic acid to be sequenced or may be any subset thereof.

The 5' and 3' ends of sequence nuclei may be overlapped to generate longer stretches of sequence. Where ambiguities arise in sequence assembly due to the availability of alternative proper overlaps with probes or sequence nuclei, hybridization with longer probes spanning the site of overlap alternatives, competitive hybridization, ligation of alternative end to end pairs of probes spanning the site of ambiguity or single pass gel analysis (to provide an unambiguous framework for sequence assembly) may be used.

By employing the above procedures, one may obtain any desired level of sequence, from a pattern of hybridization (which may be correlated with the identity of a nucleic acid sample to serve as a signature for identifying the nucleic acid sample) to overlapping or non-overlapping probes up through assembled

sequence nuclei and on to complete sequence for an intermediate fragment or an entire source DNA molecule (e.g. a chromosome).

Sequencing may generally comprise the following steps:

(a) contacting an array of immobilized oligonucleotide probes with a nucleic acid fragment under conditions effective to allow a fragment with a sequence complementary to that of an immobilized probe to form a primary complex with the immobilized probe such that the fragment has a hybridized and a non-hybridized portion;

(b) contacting a primary complex with a set of labeled oligonucleotide probes in solution under conditions effective to allow a primary complex including an unhybridized sequence complementary to that of a labeled probe to hybridize to the labeled probe, thereby forming a secondary complex wherein the fragment is hybridized with both an immobilized probe and a labeled probe;

(c) removing from a secondary complex any labeled probe that has not hybridized adjacent to an immobilized probe;

(d) detecting the presence of adjacent labeled and unlabeled probes by detecting the presence of the label; and

(e) determining a nucleotide sequence of the fragment by connecting the known sequence of the immobilized and labeled probes.

In this embodiment of SBH, ligation may be implemented by a chemical ligating agent (e.g. water-soluble carbodiimide or cyanogen bromide). A ligase enzyme, such as the commercially available T4 DNA ligase from T4 bacteriophage, may be employed. The washing conditions which are selected to distinguish between adjacent versus nonadjacent labeled and immobilized probes are selected to make use of the difference in stability of continuously stacked or ligated adjacent probes.

Numerous modifications and variations in the practice of the invention are expected to occur to those skilled in the art upon consideration of the foregoing description of the presently preferred embodiments thereof. Consequently, the only limitations which should be placed upon the scope of the present invention are those which appear in the appended claims.

Listing for Dan Coleman Fri Jun 18 14:33:11 1999 Page 1

```
#!/usr/local/bin/perl

BEGIN {
        if ( -d '/usr/local' ){
                my $length=@INC;
                foreach my $i (0 .. $length) {
                  $INC[$i]=~s/local/local/g;
                }
        }
}
use strict 'vars';
if (scalar @ARGV < 5) { die "Need ini-file, score-file, date, score-column, fi
le [FullRatio], [N] - $log:all\n"; }
my $useZeroOne=1;
my $FullRatio='inf';
$FullRatio=$ARGV[5] if ($ARGV[5]);
my $OnlyFull=0;
my $Date=$ARGV[2];
my $Colname=$ARGV[3];
if ($FullRatio eq 'inf')
{
        $OnlyFull=1;
}
$useZeroOne=$ARGV[6] if ($ARGV[6]);
my $Title = $ARGV[4];
my $MaxCandidates=5;
my $MinQuantile=.12;
my $MinRatioSum=1.0;
my $MinRatioWindowSum=.15;
my $MaxRatioRef=0.5;
my $Start=10;
my $End=-10;
my $doNotGenotype=1;
my $File='votes.$Date.$FullRatio.$useZeroOne.$Column';
my $Drivers='drivers.$Date.$FullRatio.$useZeroOne.$Column';

process_ini($ARGV[0]);

sub process_ini {
        my ($ini_file)=@_;
        open(INI,$ini_file) || die "Unable to open $ini_file\n";
        my $flag=0;
        my $type='';
        my %data;
        my $tmp;
        my $Pool;
        my $RawScores;
        while (<INI>)
        {
                chomp $_;
                if (!/^/ && $flag)
                {
                        $data{$type}{$flag}=$_;
                        $flag++;
                }
                if (/^Pools/)
                {
                        $flag=1;
                        $type='pool';
                }
```

Listing for Dan Coleman Fri Jun 18 14:33:11 1999 Page 2

```
                if (/^Pools/)
                {
                        $flag=1;
                        $type='pool';
                }
                if (/^Targets/)
                {
                        $flag=1;
                        $type='target';
                }
                if (/^Images/)
                {
                        $flag=1;
                        $type='image';
                }
                last if (/^Anything/);
        }
        die "Not enough pool info\n" if (scalar keys %{$data{'pool'}} < 2);
        foreach $flag (keys %{$data{'pool'}})
        {
                $Pool{$flag}=read_pool($data{'pool'}{$flag});
                print STDERR "Pool=$flag has ".scalar @{$Pool{$flag}}." element
\n";
        }
        #Read wild type sequence
        my ($file) = split(/\s+/,$data{'target'}{1});
        my $Seq=read_sequence($file);
        my $Scores = read_scores($ARGV[1]);
        my ($target,$pos);
        my %PoolInd;
        foreach my $pool (0..$#Pool)
        {
                foreach my $p (@{$Pool{$pool}})
                {
                        $PoolInd{$p}=$pool;
                }
        }
        my @Fill=getFills($Seq,$Pool,%PoolInd);
        foreach $target (sort { $a<=>$b } keys %{$Scores})
        {
                open (DUMP,">$File.dump.$target");
                my ($probe,$pool,$scores);
                foreach $probe (keys %{$Scores->{$target}})
                {
                        foreach $pool (keys %{$Scores->{$target}{$probe}})
                        {
                                push @scores,[new_score($probe,$pool,$Scores->{t
arget}{$probe}{$pool})];
                        }
                }
                @scores = sort { $b->[2] <=> $a->[2] } @scores;
                if ($Data eq 'best')
                {
                        my %Rank;
                        foreach my $each (0..$#scores)
                        {
                                $Rank{$scores[$each]->[0]}{$scores[$each]->[1]
}=$each;
                        }
                        my %Rank;
```

Listing for Dan Coleman — Fri Jun 18 14:32:11 1999 — Page 3

```
foreach my $pos (0..length($Seq)-1)
{
        my $lp=substr($Seq,$pos,1);
        my $pos1 = $Pos1st{substr($Seq,$pos-5,1)};
        push @Rank,$Rank{$lp}{$pos1};
}
@Rank = sort {$a<=>$b} @Rank;
my $count=0;
my $max=0;
my $bestpos=0;
my $numPulls = length($Seq)-4;
foreach my $rank (@Rank)
{
        $count++;
        my $x=$count/$numPulls-$rank/scalar @scores;
        if ($x > $max)
        {
                $max=$x;
                $bestpos=$rank;
        }
}
$Data=$bestpos+1;
@scores=@scores[0..$Data-1];
}
elsif ($Data ne 'inf')
{
        @scores=@scores[0..$Data-1];
}
my %Vote;
foreach my $score (@scores)
{
        my $Votes=1;
        $Votes = $score->[2] if ($UseScoreOnes);
        foreach my $lp (0..$Pos[$score->[1]]-1)
        {
                $Vote{$array->[1]}[$lp]+=$Votes;
        }
}
my %refScores;
my %bestScores;
my %ratios;
my %Call;
foreach $pos ($Start..length($Seq)-$End)
{
        my $theChar=substr($Seq,$pos,1);
        my $baseVote=getVotes($pos,$Seq,$Pull,\%Vote);
        my ($refScore,$bestSol,$arrRef)=sortVotes($theChar,$baseVote);

        foreach my $ch (@{$arrRef})
        {
                $bestScores{$pos}{$ch}=$baseVote->{$ch};
        }
        my $ratio = $bestSol/($refScore+.01);
        $refScores{$pos}=$refScore;
        $ratios{$pos}=$ratio;
        if ($ratio < $MaxRatioRef)
        {
                $Call{$pos}='R';
        }
        if ($ratio > $MinRatioNon)
```

Listing for Dan Coleman — Fri Jun 18 14:32:11 1999 — Page 4

```
        {
                $Call{$pos}='N';
        }
}
my @quartile = sort {$refScores{$a} <=> $refScores{$b}} keys %refScores;
foreach my $i (int($MinQuartile*@quartile)..$#quartile)
{
        my $pos=$quartile[$i];
        if (abs($ratios{$pos}-1.)<=$WindowNet)
        {
                if ($Call{$pos} eq '')
                {
                        $Call{$pos}='H';
                }
        }
}
foreach $pos (keys %ratios)
{
        if ($Call{$pos} eq '')
        {
                $Call{$pos}='X';
        }
}
my %bestChar;
foreach $pos ($Start..length($Seq)-$End)
{
        my $theChar=substr($Seq,$pos,1);
        my $baseVote=getVotes($pos,$Seq,$Pull,\%Vote);
        my ($refScore,$bestSol,$arrRef)=sortVotes($theChar,$baseVote);

        printf DUMP "%d %s %8.1f ",$pos, $theChar, $refScore;
        $bestChar{$pos}=$arrRef->[0];
        foreach my $ch (@{$arrRef})
        {
                printf DUMP "%s %8.2f ",$ch ,$baseVote->{$ch};
                $bestScores{$pos}{$ch}=$baseVote->{$ch};
        }
        my $ratio = $bestSol/($refScore+.01);
        $refScores{$pos}=$refScore;
        $ratios{$pos}=$ratio;
        printf DUMP "%5.2f %s\n",$ratio,$Call{$pos};
}
close(DUMP);
my $driver=$Driver.".$target";
open (DRIVER,">$driver");
print DRIVER 'execute\'/[analysis/NewMutScan MutScan/seqPlot.plus\'];\n';
print DRIVER 'copyPB(\'$File.dump.$target\',out=\'$File.$target.ps\',title=\'$Title Target$target\')\n';
close DRIVER;
my $dir = 'pwd';
system('/usr/loc/splus/Splus BATCH $driver log');
open(REPORT,">$File.rep.$target");
print REPORT "Title=$Title Target=$target\n";
print REPORT "Data=$Data Scale=$FullRatio used/1:$NumScoreColumns ScoreColumn:$ColumnNo\n";
print REPORT "Start=$Start End=$End\n";
print REPORT "RefThresh:$MaxRatioRef, NonThresh:$MinRatioNon, MinQuartile:$MinQuartile NetWindow:$WindowNet Candidates:$NumCandidates"
```

.../NewMutScan/seqPlot.pl

Listing for Dan Coleman
Fri Jun 16 16:33:11 1999
Page 6
Listing for Dan Coleman
Fri Jun 16 16:33:11 1999
Page 5
~ccoleMultiScanVote.pl
~ccoleMultiScanVote.pl

Listing for Dan Coleman Fri Jan 14 14:32:11 1999 Page 6

```
open (FOO, $ARGV[0]) || die "Could not read post $ARGV[0]\n";
my ($junk, $probe, $pool);
while(<FOO>)
{
    next if (/^#/);
    next if (/^$/);
    chop $_;
    ($junk,$probe,$pool)=split(/\s+/,$_);
    push @pool,$probe;
}
close(FOO);
return @pool;
}

sub read_scores {
    open(SCORE,$_[0]);
    my @scores;
    while(<SCORE>) {
        chomp;
        push(@scores,$_);
    }
    return (@scores);
}

sub new_score {
    my $n = 0;
    return $n;
}

sub portscan {
    my ($ref,$ac,$probe,$ref)=@_;
    my $refscore=$probe->{$ref};
    $ac=$ac ? $ref : $refscore-$ref;
    my $n = $ac . $refscore->{$ac} =~ $refscore->{$ac} ? keys %{$probe{$
    my $hostdir = $refscore->{$ac};
    return ($refscore,$hostdir,$n);
}

sub getname {
    my ($pool,$seq,$file,$func) = @_;
    $basefile;
    my $dir = $pool;
    $dir =~ s/\/$//;
    my ($dirname,$subdir)=$seq.$pool;
    foreach my $ch (0,2,4,7)
    {
        substr($dir,$pos,1)=$ch;
        foreach my $pool ($dir..$pool
        {
            my $probe=substr($pool,10);
            if (!$ch eq $dirchar||!$ch->{$probe})
            {
                if ($dir/$file)
                {
                    $basefile{$ch}->{$func}->{$probe};
```

Listing for Dan Coleman Fri Jan 14 14:32:11 1999 Page 7

```
my $Ref=$ref;
my $hostdir;
my $ac=0;
foreach my $pool (@pool)
{
    $hostdir=$refscore->{$pool};
    $ac=$ac ? $ref : $refscore->{$pool} ? keys %{$refscore->{$pool}} :
    $hostdir = $refscore->{$pool};
    $n=$ac ? $refscore->{$pool} : 0;
    print STDERR "$pool $n ";
    print STDERR "$n.1f ",$hostdir{$pool}{$ac}).$f
}
print STDERR "\nscore $score ";
my $dirs = $hostdir{$hostdir}->{};
my $basefile = $hostdir{$hostdir}->{};
my $filename = $basefile.$dirs.$ac;
print STDERR "$n.1f $n.1f\n",$dirs,$ac,$n,$file
}
closedir($hostdir);
    }
    closedir($dir);
}

sub read_sequence {
    my ($file)=@_;
    if ($file)
    {
        die "Unable to read wide-type sequence $file\n";
    }
    open(FH,$file);
    my @seq;
    while(<FH>)
    {
        next if (/^#/);
        next if (/^$/);
        chop $_;
        $seq.=$_;
    }
    return reff($seq);
}

sub rc {
    my $seq = shift;
    my $rseq = $seq;
    my @bases = split(//,$seq);
    my $base;
    foreach $base(@bases) $n--)
    {
        $rc .= $ref{$base}||$b;
    }
    return $rc;
}

sub read_pool {
    my ($pool,$file)=@_;
```

Listing by Dan Coleman — Fri Jun 18 14:32:11 1999 — Page 8

```
                        else
                        {
                                $baseVec{$ch}->{$name}->{$probe}=$Fullhet
ts;
                        }
                }
                foreach my $pos2 (0..3)
                {
                        next if ($pos) == ($pos-$pos2));
                        my $mutprobe=$probe;
                        foreach my $ch2 (A,C,G,T)
                        {
                                substr($mutprobe,$pos2,1)=$ch2;
                                $baseVec{$ch}->{$name}->{$mutprobe} if (
$Full->{$mutprobe} && !$onlyFull);
                        }
                }
            }
        }
        return \%baseVec;
}

sub getFulls {
        my %Full;
        my ($seq,$poolRef,$poolInd)=@_;
        foreach my $pos (0..length($seq)-10)
        {
                my $fp= substr($seq,$pos,5);
                my $lp= substr($seq,$pos+5,5);
                my $pool = $poolInd->{$lp};
                foreach $lp (@{$poolRef->[$pool]})
                {
                        $Full{$fp.$lp}=1;
                }
        }
        return %Full;
}
```

[illegible]

```
#!/usr/leo/bin/perl

BEGIN {
       if ( -d '/usr/leo'){
            my $length=$#INC;
            foreach my $i (0 .. $length) {
              $INC[$i]=-s/local/leo/g;
            }
       }
}
use strict 'vars';
if (scalar @ARGV < 6) { die "Need ini-file, score-file, #dots,
score-column, Title, DecisionRuleFile, [FullsRatio], [01 - flag:off]\n"; }

my $FullRatio="inf";
$FullRatio=$ARGV[6] if ($ARGV[6]);
my $OnlyFulls=0;
my $Dots=$ARGV[2];
my $Collumn=$ARGV[3];
if ($FullRatio eq "inf")
{
      $OnlyFulls=1;
}
my $useZeroOne=0;
$useZeroOne=$ARGV[7] if ($ARGV[7]);
my $Title = $ARGV[4];

my $NumCandidates=5;
my $MinQuartile=.12;
my $MinRatioHom=2.0;
my $MinRatioWindowHet=.15;
my $MaxRatioRef=0.5;
my $Start=0;
my $End=-1;
my $LogScale=0;
read_decision($ARGV[5]);

my $doHeterozygotes=1;
my $File="votes.$Dots.$FullRatio.$useZeroOne.$Collumn";
my $Driver="driver.$Dots.$FullRatio.$useZeroOne.$Collumn";

process_ini($ARGV[0]);

sub process_ini {
      my ($ini_file)=@_;
      open(INI,$ini_file) || die "Unable to open $ini_file\n";
      my $flag=0;
      my $type="";
      my %data;
      my @Map;
      my @Pool;
      my @RawScores;
      while (<INI>)
      {
            chop $_;
            if (!/#/ && $flag)
            {
                  $data{$type}{$flag}=$_;
                  $flag++;
            }
            if (/#Maps/)
            {
```

```
                        $flag=1;
                        $type="map";
                  }
                  if (/#Pools/)
                  {
                        $flag=1;
                        $type="pool";
                  }
                  if (/#Targets/)
                  {
                        $flag=1;
                        $type="target";
                  }
                  if (/#Images/)
                  {
                        $flag=1;
                        $type="image";
                  }
                  last if (/#Spiking/);
            }
            die "Not enough pool info\n" if (scalar keys %{$data{"pool"}} <1);
            foreach $flag (keys %{$data{"pool"}})
            {
                  $Pool[$flag]=read_pool($data{"pool"}{$flag});
                  print STDERR "Pool=$flag has ",scalar @{$Pool[$flag]},"
elements\n";
            }
            #Read wild-type sequence
            my ($file) = split(/\s+/,$data{"target"}{1});
            my $Seq=read_sequence($file);
            my $Scores = read_scores($ARGV[1]);
            my ($target,$pos);
            my %PoolInd;
            foreach my $pool (0..$#Pool)
            {
                  foreach my $lp (@{$Pool[$pool]})
                  {
                        $PoolInd{$lp}=$pool;
                  }
            }
            my $Full=getFulls($Seq,\@Pool,\%PoolInd);
            foreach $target (sort { $a<=>$b } keys %{$Scores})
            {
                  my @Names = split(/\//,$data{"target"}{$target});
                  my $name = $Names[$#Names];

                  open (DUMP,">$File.dump.$target");
                  my ($probe,$pool,@scores);
                  foreach $probe (keys %{$Scores->{$target}})
                  {
                        foreach $pool (keys %{$Scores->{$target}{$probe}})
                        {
                              push
@scores,new_score($probe,$pool,$Scores->{$target}{$probe}{$pool});
                        }
                  }
                  @scores = sort { $b->[2] <=> $a->[2] } @scores;
                  if ($Dots eq "best")
                  {
                        my %Rank;
                        foreach my $rank (0..$#scores)
                        {
```

```
$Rank{$scores[$rank]->[0]}{$scores[$rank]->[1]}=$rank;
                    }
                    my @Rank;
                    foreach my $pos (0..length($Seq)-10)
                    {
                          my $fp=substr($Seq,$pos,5);
                          my $pool = $PoolInd{substr($Seq,$pos+5,5)};
                          push @Rank,$Rank{$fp}{$pool};
                    }
                    @Rank = sort {$a<=>$b} @Rank;
                    my $cum=0;
                    my $max=0;
                    my $bestpos=0;
                    my $numFulls = length($Seq)-9;
                    foreach my $rank (@Rank)
                    {
                          $cum++;
                          my $x=($cum/$numFulls)-$rank/(scalar @scores);
                          if ($x > $max)
                          {
                                $max=$x;
                                $bestpos=$rank;
                          }
                    }
                    $Dots=$bestpos+1;
                    @scores=@scores[0..$Dots-1];
              }
              elsif ($Dots ne "inf")
              {
                    @scores=@scores[0..$Dots-1];
              }
              my %Tens;
              foreach my $score (@scores)
              {
                    my $Votes=1;
                    $Votes = $score->[2] if (!$useZeroOne);
                    $Votes = log($score->[2])/log(2) if ($LogScale &&
!$useZeroOne);
                    foreach my $lp (@{$Pool[$score->[1]]})
                    {
                          $Tens{$score->[0].$lp}=$Votes;
                    }
              }
              my %refScores;
              my %bestScores;
              my %ratios;
              my %Call;
              foreach $pos ($Start..length($Seq)+$End)
              {
                    my $theChar=substr($Seq,$pos,1);
                    my $baseVote=getVotes($pos,$Seq,$Full,\%Tens);
                    my
($refScore,$bestSol,$arrRef)=sortVotes($theChar,$baseVote);
                    foreach my $ch (@{$arrRef})
                    {
                          $bestScores{$pos}{$ch}=$baseVote->{$ch};
                    }
                    my $ratio = $bestSol/($refScore+.01);
                    $refScores{$pos}=$refScore;
                    $ratios{$pos}=$ratio;
                    if ($ratio < $MaxRatioRef)
```

```
                        {
                              $Call{$pos}="R";
                        }
                        if ($ratio > $MinRatioHom)
                        {
                              $Call{$pos}="M";
                        }
                  }
                  my @quartile = sort {$refScores{$a} <=> $refScores{$b} } keys
%refScores;
                  foreach my $i (int($MinQuartile*$#quartile)..$#quartile)
                  {
                        my $pos=$quartile[$i];
                        if (abs($ratios{$pos}-1.0)<=$MinRatioWindowHet)
                        {
                              if ($Call{$pos} eq "")
                              {
                                    $Call{$pos}="H";
                              }
                        }
                  }
                  foreach $pos (keys %ratios)
                  {
                        if ($Call{$pos} eq "")
                        {
                              $Call{$pos}="X";
                        }
                  }
                  my %bestChar;
                  foreach $pos ($Start..length($Seq)+$End)
                  {
                        my $theChar=substr($Seq,$pos,1);
                        my $baseVote=getVotes($pos,$Seq,$Full,\%Tens);
                        my
($refScore,$bestSol,$arrRef)=sortVotes($theChar,$baseVote);
                        printf DUMP "%d %s %8.3f ",$pos, $theChar, $refScore;
                        $bestChar{$pos}=$arrRef->[0];
                        foreach my $ch (@{$arrRef})
                        {
                              my $score;
                              $score= sprintf "%8.3f",$baseVote->{$ch};
                              $score="   NA   " if ($baseVote->{$ch} == -1);
                              printf DUMP "%s %s ",$ch ,$score;
                              $bestScores{$pos}{$ch}=$baseVote->{$ch};
                        }
                        my $ratio = $bestSol/($refScore+.01);
                        $refScores{$pos}=$refScore;
                        $ratios{$pos}=$ratio;
                        printf DUMP "%5.3f %s\n",$ratio,$Call{$pos};
                  }
                  close(DUMP);
                  my $driver=$Driver.".$target";
                  open (DRIVER,">$driver");
                  print DRIVER
"source(\"/f3analysis/NewMutScan/MutScan/seqPlot.splus\")\n";
                  print DRIVER
"coryPS(\"$File.dump.$target\",out=\"$File.$target.ps\",tit=\"$Title
Targ=$target $name\")\n";
                  close DRIVER;
                  my $dir = `pwd`;
                  system("/usr/leo/splus/Splus BATCH $driver log");
                  open(REPORT,">$File.rep.$target");
```

```
            print REPORT "Title=$Title Target=$target\n";
            print REPORT "Dots=$Dots Scale=$FullRatio use0/1:$useZeroOne
ScoreCollumn:$Collumn\n";
            print REPORT "Start=$Start End=$End\n";
            print REPORT "RefThresh:$MaxRatioRef, HomThresh:$MinRatioHom,
HetQuartile:$MinQuartile HetWindow:$MinRatioWindowHet
Candidates:$NumCandidates\n";
            my %count;
            foreach $pos ($Start..length($Seq)+$End)
            {
                  $count{$Call{$pos}}++;
                  if (($Call{$pos} eq "M")||($Call{$pos} eq "H"))
                  {
                        print REPORT "$Call{$pos} $pos
",substr($Seq,$pos,1),"-",$bestChar{$pos},"\n";
                  }
            }
            print REPORT "---------------\nType # %\n";
            foreach my $call (R,M,H,X)
            {
                  printf REPORT "%s %3d
%5.3f\n",$call,$count{$call}+0,$count{$call}/(scalar keys %Call);
            }
            close(REPORT);
            my @pos = sort { $ratios{$b}<=>$ratios{$a} } (keys %ratios);
            my @poses;
            foreach my $i (0..$NumCandidates-1)
            {
                  push @poses,$pos[$i] if
($ratios{$pos[$i]}>=$MinRatioHom);
            }
            print STDERR  "T=$target Examining ",scalar @poses," homozygote
mutations at bases @poses\n";
            open(HOMO,">$File.hom.$target");
            print HOMO "pos char ratio refRegion bestRegion discrim
discrimRatio refRatio\n";
            foreach $pos (@poses)
            {
                  my $min = $pos-9;
                  $min = 0 if ($min<0);
                  my $max = $pos+9;
                  $max = length($Seq)-10 if ($max>(length($Seq)-10));
                  my $votesRef=0;
                  my $votesBest=0;
                  foreach my $pos2 ($min..$max)
                  {
                        my $weight = 1-abs($pos2-$pos)*.1;
                        $votesRef+=$refScores{$pos2}*$weight;
                        my @best = sort
{$bestScores{$pos2}{$b}<=>$bestScores{$pos2}{$a}} keys
%{$bestScores{$pos2}};
                        my $best = $bestScores{$pos2}{$best[0]};
                        $votesBest+=$best*$weight;
                  }
                  my $refDiscrim = $votesRef/$votesBest;
                  my $theChar = substr($Seq,$pos,1);
                  foreach my $Ch (A,C,T,G)
                  {
                        my $mut=$Seq;
                        substr($mut,$pos,1)=$Ch;
                        my $Full2=getFulls($mut,\@Pool,\%PoolInd);
                        my $mutVotes=0;
```

```
                        my $mutBest=0;
                        foreach my $pos2 ($min..$max)
                        {
                              my
$baseVote=getVotes($pos2,$mut,$Full2,\%Tens);
                              my
($refScore,$bestSol,$arrRef)=sortVotes(substr($mut,$pos2,1),$baseVote);
                              my $weight = 1-abs($pos2-$pos)*.1;
                              $mutVotes+=$refScore*$weight;
                              $mutBest+=$bestSol*$weight;
                        }
                        print HOMO "$pos $Ch ";
                        printf HOMO "%5.3f
",$bestScores{$pos}{$Ch}/($refScores{$pos}+.01);
                        printf HOMO "%8.3f %8.3f ",$mutVotes,$mutBest;
                        my $discrim = $mutVotes/($mutBest+.01);
                        my $ratioToRef = $mutVotes/($votesRef+.01);
                        my $discrimRatio=$discrim/($refDiscrim+.01);
                        printf HOMO "%5.3f %5.3f
%5.3f\n",$discrim,$discrimRatio,$ratioToRef;
                  }

            }
            close(HOMO);
            next if (!$doHeterozygotes);
            #examine heterozygotes
            my @refs = sort {$refScores{$a}<=>$refScores{$b}} keys
%refScores;
            my @poses;
            my @pose;
            foreach my $i (int($MinQuartile*$#refs)..$#refs)
            {
                  my $pos=$refs[$i];
                  push @pose,$pos if
(abs($ratios{$pos}-1.0)<=$MinRatioWindowHet);
            }
            @poses = sort { abs($ratios{$a}-1.0) <=> abs($ratios{$b}-1.0) }
@pose;
            my $min=$#poses;
            $min=$NumCandidates-1 if ($NumCandidates <= $min);
            @poses = @poses[0..$min];
            print STDERR "T=$target Examining ",scalar @poses,"
heterozygote mutations at bases @poses\n";
            open(HETER,">$File.het.$target");
            print HETER "pos char ratio refRegion bestRegion discrim
discrimRatio refRatio\n";
            foreach $pos (@poses)
            {
                  my $min = $pos-9;
                  $min = 0 if ($min<0);
                  my $max = $pos+9;
                  $max = length($Seq)-10 if ($max>(length($Seq)-10));
                  my $votesRef=0;
                  my $votesBest=0;
                  foreach my $pos2 ($min..$max)
                  {
                        my $weight = 1-abs($pos2-$pos)*.1;
                        $votesRef+=$refScores{$pos2}*$weight;
                        my @best = sort
{$bestScores{$pos2}{$b}<=>$bestScores{$pos2}{$a}} keys
%{$bestScores{$pos2}};
                        my $best = $bestScores{$pos2}{$best[0]};
```

```
                        $votesBest+=$best*$weight;
                    }
                    my $refDiscrim = $votesRef/$votesBest;
                    my $theChar = substr($Seq,$pos,1);
                    foreach my $Ch (A,C,T,G)
                    {
                        my $mut=$Seq;
                        substr($mut,$pos,1)=$Ch;
                        my $Full2=getFulls($mut,\@Pool,\%PoolInd);
                        my $mutVotes=0;
                        my $mutBest=0;
                        foreach my $pos2 ($min..$max)
                        {
                            my
$baseVote=getVotes($pos2,$mut,$Full2,\%Tens);
                            my
($refScore,$bestSol,$arrRef)=sortVotes(substr($mut,$pos2,1),$baseVote);
                            my $weight = 1-abs($pos2-$pos)*.1;
                            $mutVotes+=$refScore*$weight;
                            $mutBest+=$bestSol*$weight;
                        }
                        print HETER "$pos $Ch ";
                        printf HETER "%5.3f
",$bestScores{$pos}{$Ch}/($refScores{$pos}+.01);
                        printf HETER "%8.3f %8.3f ",$mutVotes,$mutBest;
                        my $discrim = $mutVotes/($mutBest+.01);
                        my $ratioToRef = $mutVotes/($votesRef+.01);
                        my $discrimRatio=$discrim/($refDiscrim+.01);
                        printf HETER "%5.3f %5.3f
%5.3f\n",$discrim,$discrimRatio,$ratioToRef;
                    }
                }
                close(HETER);
        }
}

sub read_sequence {
        my ($file)=@_;
        if (!-e $file)
        {
                die "Unable to read wild-type sequence $file\n";
        }
        open(WT,$file);
        my $Seq;
        while (<WT>)
        {
                next if (/#/);
                next if (/\+/);
                chop $_;
                $Seq.=uc($_);
        }
        return rc($Seq);
}

sub rc {
        my $seq = shift;
        my %rc = ( "A","T", "T","A", "C","G", "G","C" );
        my @chars = split(//,$seq);
        my $rc="";
        for(my $i=$#chars; $i>=0; $i--)
        {
                $rc .= $rc{$chars[$i]};
```

```
        }
        return $rc;
}

sub read_pool {
        my ($poolfile)=@_;
        open (POOL,$poolfile) || die "Could not read pool $poolfile\n";
        my ($junk, $probe, @pool);
        while(<POOL>)
        {
                next if (/>/);
                next if (/Group/);
                chop $_;
                ($junk,$probe)=split(/\s+/,$_);
                push @pool,$probe;
        }
        close(POOL);
        return \@pool;
}

sub read_scores {
        my $file=shift;
        open(SCORE,$file);
        my %Scores;
        while(<SCORE>)
        {
                chop $_;
                my($target,$probe,$pool,@scores)=split(/\s+/,$_);
                $Scores{$target}{$probe}{$pool}=$scores[$Collumn];
        }
        return (\%Scores);
}

sub new_score {
        my @sc = @_;
        return \@sc;
}

sub sortVotes {
        my ($refChar,$voteRef)=@_;
        my $refScore=$voteRef->{$refChar};
        delete $voteRef->{$refChar};
        my @ch = sort { $voteRef->{$b} <=> $voteRef->{$a} } keys %{$voteRef};
        my $bestSol = $voteRef->{$ch[0]};
        my @subs;
        foreach my $ch (@ch)
        {
                if ($ch eq uc($ch))
                {
                        push @subs,$ch;
                }
        }
        foreach my $ch (@ch)
        {
                if ($ch eq lc($ch))
                {
                        push @subs,$ch;
                }
        }
        pop @subs;
        return ($refScore,$bestSol,\@subs);
}
```

```
sub getVotes {
      my ($pos,$seq,$Full,$Tens) = @_;
      my %baseVote;
      my $min = $pos-9;
      $min=0 if ($min<0);
      my $max = $pos;
      $max = length($seq)-10 if ($max > (length($seq)-10));
      my $theChar=substr($seq,$pos,1);
      foreach my $ch (A,C,G,T,a,c,g,t,d)
      {
            my $mut=$seq;
            if ($ch eq uc($ch)) {
                  substr($mut,$pos,1)=$ch;
            }
            elsif ($ch eq "d")  {
                  substr($mut,$pos,1)="";
                  if (substr($seq,$pos,1) eq substr($seq,$pos+1,1))
                  {
                        $baseVote{$ch}=-1;
                        next;
                  }
            }
            else  {
                  substr($mut,$pos,0)=uc($ch);
                  if (substr($seq,$pos,1) eq uc($ch))
                  {
                        $baseVote{$ch}=-2;
                        next;
                  }
            }
            $baseVote{$ch}=0.0;
            foreach my $pos2 ($min..$max)
            {
                  my $probe=substr($mut,$pos2,10);
                  if (($ch eq $theChar)||(!$Full->{$probe}))
                  {
                        if ($OnlyFulls)
                        {
                              $baseVote{$ch}+=$Tens->{$probe};
                        }
                        else
                        {
                              $baseVote{$ch}+=$Tens->{$probe}*$FullRatio;
                        }
                  }
                  next if ($OnlyFulls);
                  foreach my $pos3 (0..9)
                  {
                        next if ($pos3 == ($pos-$pos2));
                        my $mutprobe=$probe;
                        foreach my $ch2 (A,C,G,T)
                        {
                              substr($mutprobe,$pos3,1)=$ch2;
                              $baseVote{$ch}+=$Tens->{$mutprobe} if
(!$Full->{$mutprobe});
                        }
                  }
            }
      }
      return \%baseVote;
}
```

```
sub getFulls {
      my %Full;
      my ($seq,$poolRef,$pcolInd)=@_;
      foreach my $pos (0..length($seq)-10)
      {
            my $fp= substr($seq,$pos,5);
            my $lp= substr($seq,$pos+5,5);
            my $pool = $poolInd->{$lp};
            foreach $lp (@{$poolRef->[$pool]})
            {
                  $Full{$fp.$lp}=1;
            }
      }
      return \%Full;
}

sub read_decision {
      my $file = shift;
      open (DEC,$file) || die "Unable to open decision rule file:$file\n";
      while (<DEC>)
      {
            chop $_;
            my @fields = split(/\s+/,$_);
            if ($fields[0] =~ /log/)
            {
                  $LogScale = $fields[1];
            }
            if ($fields[0] =~ /candidates/)
            {
                  $NumCandidates=$fields[1];
            }
            if ($fields[0] =~ /quartile/)
            {
                  $MinQuartile=$fields[1];
            }
            if ($fields[0] =~ /window/)
            {
                  $MinRatioWindowHet=$fields[1];
            }
            if ($fields[0] =~ /ref/)
            {
                  $MaxRatioRef=$fields[1];
            }
            if ($fields[0] =~ /hom/)
            {
                  $MinRatioHom=$fields[1];
            }
            if ($fields[0] =~ /start/)
            {
                  $Start=$fields[1];
            }
            if ($fields[0] =~ /end/)
            {
                  $End=$fields[1];
            }
      }
}
```

WHAT IS CLAIMED IS:

1. A method of sequencing a target nucleic acid, comprising:

(a) contacting a target nucleic acid with a plurality of oligonucleotide probes of predetermined length and predetermined sequence, wherein each probe comprises an information region, under conditions which produce, on average, relatively more probe:target hybrids per probe for probes that are perfectly complementary in the information region of the probe than for probes that are substantially perfectly complementary in the information region of the probe, and relatively fewer probe:target hybrids that are significantly mismatched in the information region of the probe;

(b) measuring the hybridization signal of said probes with the target nucleic acid; and

(c) assigning a numerical voting score to each probe or pool of probes based on the relative strength of hybridization signal; and

(d) determining the sequence of the target nucleic acid, comprising the step of summing the numerical voting scores of the probes in relation to their sequences.

2. The method of claim 1 further comprising the step of modifying said numerical voting score assigned in step (c) by a voting factor selected based on the relationship of the probe to hypothetical target nucleic acid sequence.

3. A method of sequencing a target nucleic acid, comprising:

(a) contacting a target nucleic acid with two sets of a plurality of oligonucleotide probes of predetermined length and predetermined sequence, wherein each probe comprises an information region, and wherein one set of the probes are attached to a fixed support, under conditions which produce, on average, relatively more probe:target hybrids per probe for probes that are perfectly complementary in the information region of the probe than for probes

that are substantially perfectly complementary in the information region of the probe, and relatively fewer probe:target hybrids that are significantly mismatched in the information region of the probe;

(b) covalently joining probes that form contiguous probe:target hybrids that are capable of ligation;

(c) measuring the hybridization signal of said covalently joined probes with the target nucleic acid; and

(d) assigning a numerical voting score to each covalently joined probe or pool of covalently joined probes based on the relative strength of hybridization signal; and

(e) determining the sequence of the target nucleic acid, comprising the step of summing the numerical voting scores of the probes in relation to their sequences.

4. The method of claim 3 further comprising the step of modifying said numerical voting score assigned in step (d) by a voting factor selected based on the relationship of the probe to hypothetical target nucleic acid sequence.

5. The method of claim 3 wherein the probes not attached to a fixed support are labeled, and wherein the label is detected.

6. The method of claim 1 or 3 wherein the probes are divided into pools or subpools and all probes within each pool or subpool are associated with an identification tag unique to the pool or subpool.

7. The method of claim 6 wherein the identification tag is a fluorescent label.

8. An improved method of sequencing by hybridization using subsets of probes that have been labeled with different tags and pooled in one or more pools by mixing all or a number of probes from each subset.

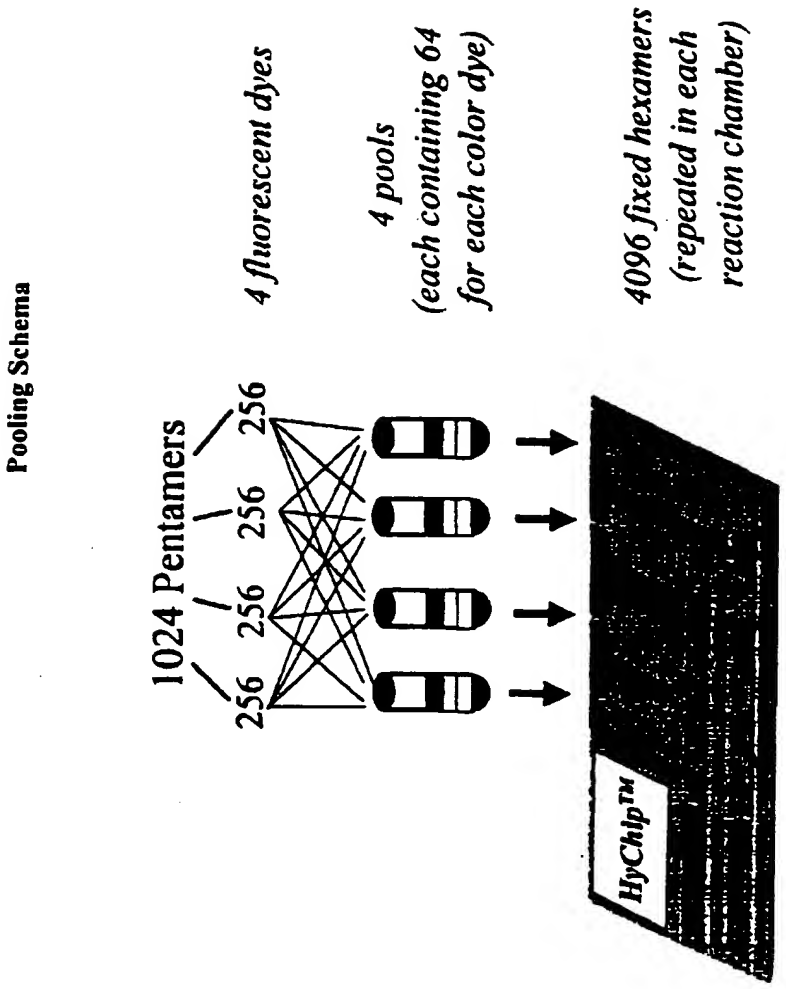
Pooling Schema
1024 Pentamers
256
256
256
256
4 fluorescent dyes
4 pools
(each containing 64
for each color dye)
4096 fixed hexamers
(repeated in each
reaction chamber)
HyChip™

FIG. 1

Sequencing of 700 bp fragment of human apo-B gene

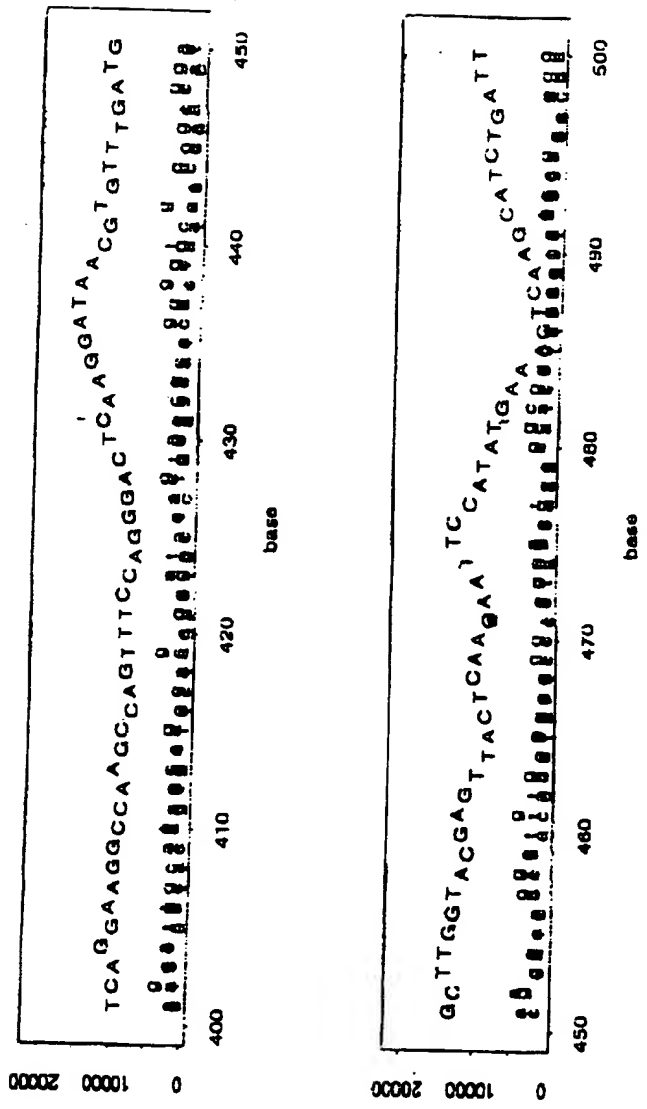

FIG. 2

INTERNATIONAL SEARCH REPORT

Interna ıl Application No
PCT/US 00/16899

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 C12Q1/68

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 C12Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
E	WO 00 40758 A (HYSEQ INC) 13 July 2000 (2000-07-13) claims 1-35 ---	1-8
A	WO 98 31836 A (HYSEQ INC) 23 July 1998 (1998-07-23) claims 1-4,7-25,29-33 ---	1-8
A	US 5 695 940 A (CRKVENJAKOV RADOMIR B ET AL) 9 December 1997 (1997-12-09) claims 1-3 ---	1-8
A	WO 95 09248 A (ARCH DEV CORP ;DRMANAC RADOJE (US)) 6 April 1995 (1995-04-06) cited in the application the whole document --- -/--	1-8

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 October 2000	31/10/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Osborne, H

2

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Interna[illegible]l Application No
PCT/US 00/16899

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 717 113 A (AFFYMAX TECH NV) 19 June 1996 (1996-06-19) -----	

2

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

...mation on patent family members

Interna ...l Application No
PCT/US 00/16899

Patent document cited in search report		Publication date	Patent family member(s)			Publication date
WO 0040758	A	13-07-2000	NONE			
WO 9831836	A	23-07-1998	AU	6131798	A	07-08-1998
			CN	1250485	T	12-04-2000
			EP	0968305	A	05-01-2000
US 5695940	A	09-12-1997	US	5525464	A	11-06-1996
			YU	57087	A	31-08-1990
			US	5972619	A	26-10-1999
			US	5492806	A	20-02-1996
			US	5667972	A	16-09-1997
			US	5202231	A	13-04-1993
			US	6018041	A	25-01-2000
WO 9509248	A	06-04-1995	AU	694146	B	16-07-1998
			AU	8072794	A	18-04-1995
			BR	9407712	A	12-02-1997
			CA	2172722	A	06-04-1995
			CN	1136330	A	20-11-1996
			CZ	9600905	A	16-10-1996
			EP	0723598	A	31-07-1996
			FI	961283	A	22-05-1996
			HU	75993	A	28-05-1997
			JP	9505729	T	10-06-1997
			NO	961165	A	23-05-1996
			NZ	275194	A	22-09-1997
			PL	313735	A	22-07-1996
			RU	2143004	C	20-12-1999
EP 0717113	A	19-06-1996	US	5795716	A	18-08-1998
			US	5974164	A	26-10-1999